Ce este şi cum funcţioneaza poşta electronică

Înca din cele mai vechi timpuri oamenii si-au pus problema comunicarii. În istorie, cea mai primitivă modalitate de comunicare a fost cea prin foc. În perioada invaziilor turcesti, oamenii dintr-un sat semnalau acest eveniment, catre localnicii satelor vecine, prin aprinderea unui foc pe o colina. Aceasta modalitate de comunicare ne face (pe noi, cei care stam în fata unui calculator si prin simpla miscare a degetelor sa comunicam cu o persoana din orice colt al lumii) sa zâmbim. Dar fiecare epoca cu problemele ei ... razesii lui Stefan cel Mare se bateau cu turcii, administratorii din ziua de azi se lupta cu hackerii. Astăzi subiectul poştă electronică e… fierbinte.

Serviciul de poştă electronica (e-mail - electronic mail) a stat la baza dezvoltării Internet-ului, constituind o modalitate de a satisface nevoia de comunicare, permiţând trimiterea de documente electronice între utilizatorii retelei. Toţi autorii din literatura de specialitate au fost de acord că: poşta electronică a fost calul troian al tehnologiilor groupware.

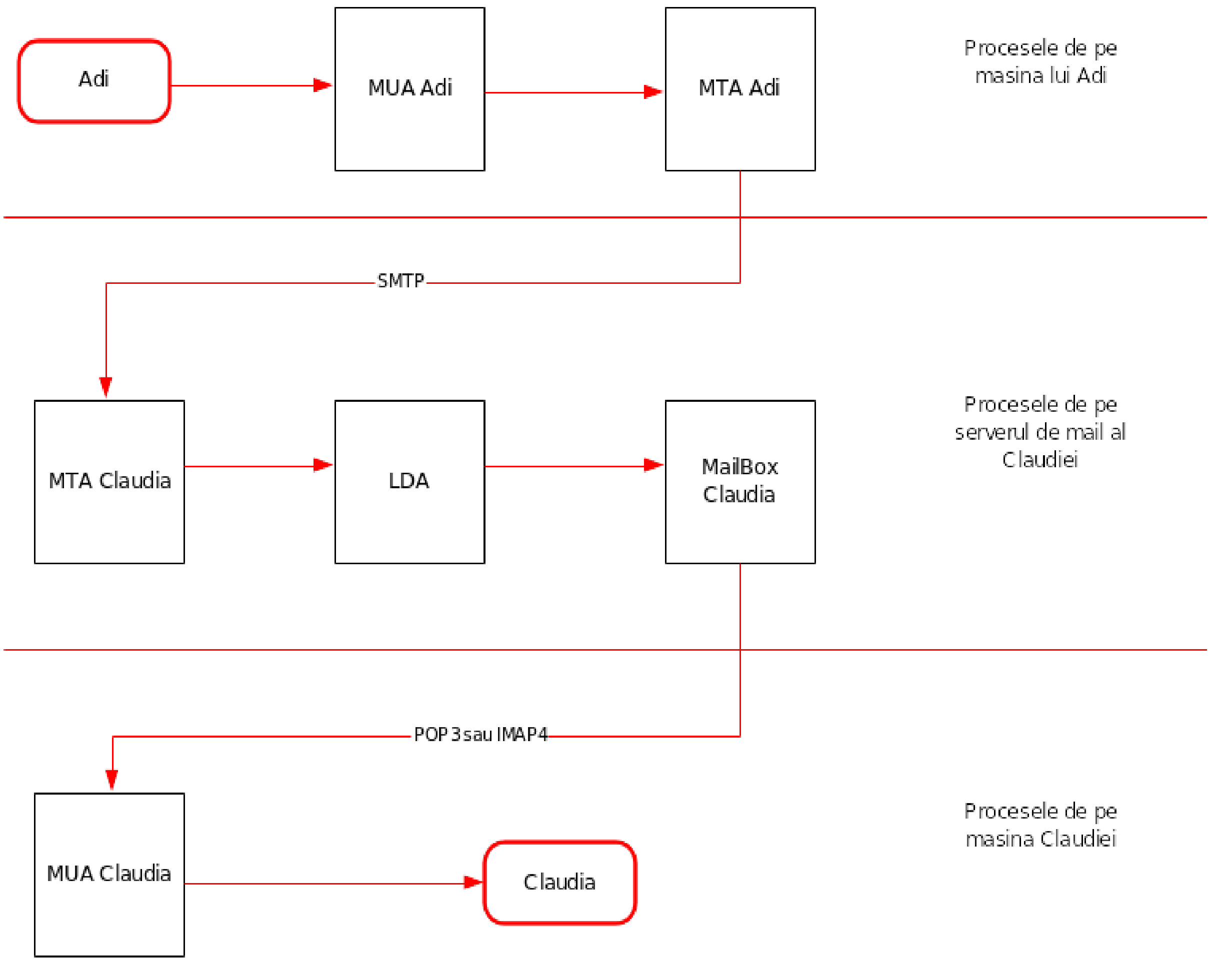
Vorbim de trei componente esenţiale E-mail

- agentul utilizator - (***MUA*** *– Mail User Agent*) - un program cu care utilizatorul îsi citeste si trimite posta electronica,
- server-ul de posta electronica (cutia postala) - locul în care ajunge e-mail-ul si din care agentul utilizator preia mesajele sosite,
- agentii de transfer postal (***MTA*** *- Mail Transfer Agent*) - un program care preia mesajele de la MUA si le transmit catre cutia postala a destinatarului.

Sa vedem care este "drumul" pe care îl urmeaza un mesaj, din momentul în care expeditorul apasa pe butonul "*Send*" al programului sau de e-mail si pâna când ajunge în casuta postala a destinatarului. Agentul utilizator al expeditorului plaseaza mesajul într-o coada prelucrata de agentul de transfer care o trimite apoi spre destinatar. În acest caz, programul *MTA* actioneaza ca un client si contacteaza server-ul masinii de la distanta, pe care se afla contul (cutia postala) a destinatarului. Clientul stabileste o legatura *TCP* cu server-ul si îi trimite mesajul, în conformitate cu protocolul *SMTP* (*Simple Mail Transport Protocol*) si care "asculta" pe portul 25. Server-ul primeste mesajul si îl plaseaza în cutia destinatarului.

Nu întotdeauna programul *MTA*, contactat de *MUA*, reuseste sa stabileasca o legatura directa cu host-ul pe care se afla contul destinatie. De cele mai multe ori *MTA* trimite mesajul unui alt program *MTA*, aflat pe un calculator "mai apropiat" de calculatorul destinatie. Observam ca

programul *MTA* trebuie sa fie atât program client cât si server. El trebuie sa stie cum sa trimita un mesaj dar si cum sa primeasca un mesaj din partea altui *MTA*.



Protocolul SMTP

Protocolul SMTP asigura o comunicatie bidirectionala între programele MTA client si server. Clientul MTA trimite comenzi catre server-ul MTA care, la rândul sau, trimite raspunsuri clientului. Cu alte cuvinte, comenzile SMTP necesita raspunsuri de la modulul SMTP receptor.

I. **Comenzi SMTP.** Pot fi definite ca fiind siruri de caractere terminate cu caracterul (Carriage-Return Line-Feed). Pentru a întelege mai usor comenzile SMTP vom apela la analiza unui exemplu de tranzactie de posta. Presupunem ca utilizatorul Ut, de pe host-ul "hostT.ro" trimite un mesaj catre utilizatorii Ur1, Ur2, Ur3, cu conturi pe calculatorul "hostR.ro". Pentru a simplifica explicarea procesului vom numerota fiecare linie, trecând în dreptul ei caracterul "R" sau "T", functie de emitentul liniei: Receptorul sau Transmitatorul.

```
1.  R:    220 hostR.ro Simple Mail Transfer Service Ready
2.  T:    HELO hostT.ro
3.  R:    250 hostR.ro
4.  T:    MAIL FROM:
5.  R:    250 OK
6.  T:    RCPT TO:
7.  R:    250 OK
8.  T:    RCPT TO:
9.  R:    550 No such user here
10. T:    RCPT TO:
11. R:    250 OK
12. T:    DATA
13. R:    354 Start mail input; end with
 .
14. T:    Una, alta, …….. 
15. T:    ….. etc, etc, etc, ……
16. T:    .
17. R:    250 OK
18. T:    QUIT
19. R:    221 hostR.ro Service closing transmission channel
```

Când un program *MTA* stabileste o conexiune *TCP* cu un alt *MTA*, acesta din urma raspunde cu codul 220, ceea ce înseamna ca serviciile de posta electronica, de pe calculatorul contactat, sunt disponibile (linia 1). Dupa primirea codului de raspuns 220, conform specificatiilor *SMTP*, clientul *MTA* trebuie sa trimita comanda *HELO*. Asa cum se vede si din linia 2, comanda *HELO* necesita ca argument hostul client. Serverul *MTA* receptioneaza comanda *HELO* si trimite codul de raspuns 250. Acest cod anunta transmitatorul ca actiunea solicitata a fost încheiata cu succes (linia 3).

În urma acestei "discutii introductive", clientul *MTA* va initia tranzactia mail. Pentru aceasta *MTA* poate transmite una din urmatoarele comenzi: *MAIL*, *SEND*, *SOML*, *SAML*. Asa cum se vede din linia 4, în cazul nostru este folosita comanda MAIL (celelalte trei comanzi sunt considerate optionale si de aceea sunt foarte rar implementate in programe *MTA*). Ca urmare a acestei comenzi, serverul va trimite din nou un cod de raspuns 250 care va indica faptul ca adresa cutiei postale a transmitatorului este adecvata (linia 5).

În continuare, clientul *MTA* va transmite comanda *RCPT* (pentru a identifica un anumit destinatar al mail-ului), asa cum se vede din liniile 6, 8, 10. În liniile 7 si 11, serverul *MTA* transmite un cod 250, ceea ce înseamna ca destinatarul specificat în comanda RCPT are cutie postala pe host-ul respectiv. În linia 9 se observa receptionarea unui cod de raspuns 550, ceea ce arata ca *MTA* nu poate îndeplini cererea clientului.

Primind raspuns la toate comenzile RCPT, clientul trebuie sa trimita o comanda *DATA* pentru a comunica serverului ca va trimite datele mail. Linia 12 prezinta comanda *DATA*, iar linia 13 mesajul serverului cu codul de raspuns 354. Acest cod de raspuns spune clientului *MTA* sa initieze transferul datelor si sa semnalizeze sfârsitul lor cu o secventa de caractere . (o linie noua care contine doar un punct). În acest moment începe transmiterea datelor (liniile 14, 15, 16). Dupa primirea secventei, serverul va raspunde cu codul 250 (linia 17). Pentru a încheia tranzactia mail, *SMTP* cere clientului *MTA* sa transmita o comanda *QUIT* (linia 18), fapt pentru care serverul va reactiona prin codul de raspuns 221 - accepta cererea de închidere a canalului de transmisie.

În orice moment al tranzactiei mail, clientul poate utiliza o gama de comenzi optionale, cum ar fi:

- NOOP (NO OPeration)- cere serverului MTA sa nu execute alta actiune decât returnarea unui raspuns OK,
- HELP - cere serverului MTA sa transmita informatii suplimentare clientului,
- VRFY - cere serverului sa verifice daca argumentul identifica un utilizator valid.

II. **Raspunsuri SMTP**. Serverele MTA raspund fiecarei comenzi SMTP cu un cod de raspuns format din trei cifre urmat de informatie text auxiliara. Exista cazuri când un singur cod de raspuns poate include mai multe linii de text. Fiecare cifra din codul de raspuns SMTP are o semnificatie speciala:

- Prima cifra poate lua cinci valori: 1, 2, 3, 4, 5 - cifra 1 indica faptul ca serverul MTA a accepta comanda, dar actiunea solicitata cere clientului MTA sa confirme informatia de raspuns. Clientul va trebui sa transmita o alta comanda care sa specifice daca actiunea se continua sau se abandoneaza. În implementarile actuale SMTP nu contine comenzi care sa permita acest gen de raspuns, el a fost inclus în specificatiile SMTP pentru posibile comenzi viitoare. Aceasta poate fi considerata o planificare în avans. - cifra 2 indica faptul ca serverul MTA a încheiat cu succes actiunea solicitata si clientul poate initia o noua comanda. - cifra 3 arata ca serverul MTA accepta comanda, dar cere informatii suplimentare pentru a executa actiunea solicitata. - cifra 4 indica faptul ca serverul nu accepta comanda, deci actiunea solicitata nu este îndeplinita. În acest caz eroarea este temporara si clientul poate sa încerce din nou. - cifra 5 are aproape aceeasi semnificatie cu cifra 4 cu deosebirea ca eroarea este mai grava si clientul nu trebuie sa mai repete aceeasi cerere.
- Cifra a doua precizeaza categoriile de erori: - cifra 0 identifica erorile de sintaxa, - cifra 1 desemneaza raspunsuri de informatie, - cifra 2 mentioneaza canalul de transmisie, - cifra 5 specifica raspunsuri legate de însusi sistemul de mail. În mod curent, SMTP nu aloca coduri de raspuns cu a doua cifra 3 sau 4.
- Cifra a trei-a are rol, doar, de numerotare a codurilor de raspuns diferite, dintr-o anumita serie. De exemplu, codurile de raspuns 500-

504 sunt toate, mesaje de eroare de sintaxa. Cifra a trei-a are rol de a diferentia, între ele, mesajele din aceeasi categorie.

În principiu, sunt **trei tipuri de acces la cutia postala**, functie de serviciile oferite de ISP-ul care gazduieste contul utilizator:

- Prin acces direct sau "Remote Acces" la serverul e-mail. Aceasta varianta este mai rar utilizata din cauza strategiilor de securitate impuse de ISP,
- Prin utilizarea unor protocoale specializate, cum ar fi POP3 si IMAP,
- Acces prin interfata web. Aceasta modalitate capata o pondere din ce în e mai mare, datorita "exploziei" care o cunoaste www-ul an momentul actual.

Protocolul POP

Protocolul POP (*Post Office Protocol*) functioneaza foarte asemanator cu protocolul *SMTP*. În prezent, exista doua versiuni: *POP2* si *POP3*. Cu toate ca suntem tentati, la prima vedere, sa credem ca POP3 este o revizuire (îmbunatatire) o protocolului POP2, cele doua protocoale sunt total diferite, folosind chiar porturi de comunicatie diferite. POP2 este mult mai apropiat de *SMTP* decât *POP3*, comenzile si structura lor fiind mult mai apropiate de comenzile *SMTP*.

Protocolul POP3 defineste trei stadii distincte prin care poate trece o sesiune de lucru: autorizare, tranzactie si actualizare. În starea de autorizare clientul trebuie sa se autentifice pe server (nume utilizator + parola). Daca aceasta etapa s-a încheiat cu succes, serverul deschide cutia postala a clientului si sesiunea trece în starea de tranzactie. În aceasta stare, clientul poate cere serverului sa-i ofere anumite date (o lista a mesajelor) sau sa efectueze o anumita actiune (preluarea mesajelor). Când serverul termina de executat comenzile clientului, sesiunea POP3 intra în starea actualizare si conexiunea se închide.

Prezentam mai jos comenzile POP3 necesare pentru o implementare minimala a acestui protocol pe Internet.

Comanda	**Descriere**
USER	Cere un nume care identifica utilizatorul
PASS	Cere o parola pentru utilizator/server
QUIT	Închide conexiunea TCP
STAT	Serverul returneaza numarul de mesaje din cutia postala si dimensiunea totala a mesajelor
LIST	Returneaza ID-urile si dimensiunile mesajelor, afisate linie cu linie (permite un ID ca parametru, caz în care returneaza dimensiunea mesajului identificat prin ID-ul respectiv)
RETR	Preia un mesaj din cutia postala (Necesita un ID de mesaj ca parametru)
DELE	Marcheaza un mesaj pentru stergere (Necesita un ID de mesaj ca parametru)

NOOP	Serverul returneaza un raspuns pozitiv, dar nu executa nici o actiune
LAST	Serverul returneaza cel mai mare numar de mesaj care a fost accesat
RSET	Deselecteaza toate mesajele marcate pentru stergere

Chiar daca acest protocol defineste mai multe comenzi, contine doar doua posibilitati de raspuns:

1. *+OK* - folosit pentru un raspuns pozitiv (analog cu *ACK* - de confirmare)
2. -*ERR* - folosit pentru un raspuns negativ (analog cu *NAK* - operatie esuata)

Se poate considera ca ambele raspunsuri "au succes", în sensul ca serverul POP3 a receptionat comanda si a returnat un raspuns.

Protocolul IMAP - (Interactive Mail Access Protocol)

Este un protocol ceva mai complex decât *POP3*, proiectat special pentru utilizatorii care nu-si acceseaza posta, tot timpul, de pe acelasi calculator (ex: de pe un PC la serviciu si alt PC acasa).

Ideea de baza, de la care s-a pornit proiectarea acestui protocol, este ca serverul sa pastreze un "depozit" central de mesaje, accesat de pe orice alt calculator. Cu alte cuvinte *IMAP* nu face "download" la mesaje, ci le lasa pe server (ca si în cazul programelor *MUA* bazate pe interfata *WEB*).

Un alt avantaj IMAP, este posibilitatea accesarii unui mesaj utilizând atribute (Ex: da-mi primul mesaj de la Cristi). Spre deosebire protocolul *POP3* permite accesarea mesajelor doar dupa *ID*-ul lor. Utilizând protocolul *IMAP*, o csutie postala poate fi comparata mai degraba cu un sistem de baze de date, decât cu o secventa liniara de mesaje.

Confidentialitatea postei electronice

Un mesaj, în drumul sau de la expeditor la destinatar, poate trece pe la un numar foarte mare de calculatoare. Oricare dintre acestea poate sa-l citeasca si/sau sa faca o copie a lui, dreptul la intimitate al mesagerii electronice fiind doar o poveste frumoasa. Acest neajuns al suitei de protocoale *TCP/IP* a "pus la treaba" mai multe persoane si grupuri de utilizatori care si-au pus problema confidentialitatii mesajelor. Astfel au luat nastere mai multe sisteme de criptare a mesajelor, cum ar fi *PGP* si *PEM*.

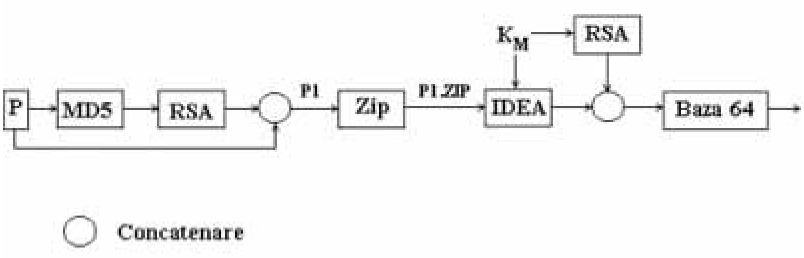
PGP - Pretty Good Privacy (confidentialitate destul de buna)

Acest sistem de criptare are drept initiator pe *Phil Zimmerman* si este un pachet complet de securitate a e-mail-ului, asigurând confidentialitate, autentificare, semnatura digitala si compresie. Sistemul este disponibil pe mai toate Sistemele de Operare, fiind distribuit pe Internet, în mod

gratuit (inclusiv toate sursele). Din acest motiv a creat si numeroase controverse. Guvernul Statelor Unite a condamnat PGP ca violeaza legea exportului de munitie, fiind usor disponibil pentru utilizatorii din afara S.U.A.. Aceasta restrictie impusa de guvernul S.U.A. a fost rezolvata prin dezvoltarea versiunilor ulterioare în afara Statelor Unite. Dar hartuiala nu s-a terminat. Multe institutii nu se împaca cu ideea ca prin Internet doi utilizatori pot schimba mesaje fara sa poata fi "supravegheati". Cu toate acestea motto-ul lui *Zimmerman* nu s-a schimbat: "*daca dreptul la confidentialitate este în afara legii, atunci doar cei aflati în afara legii vor beneficia de confidentialitate*".

Din aceste motive PGP se bazeaza, în principal, pe algoritmi (*RSA*, *IDEA*, *MD5*) care au fost minutios analizati si nu au fost proiectati sau influentati de vreo organizatie guvernamentala, pentru a lasa unele "portite".

În figura de mai jos prezentam schematic acest sistem:



Utilizatorul A trimite lui B un mesaj simplu (notat P). A si B au cheile private (*DX*) si cheile *RSA* publice (*EX*).

Prin executarea programului *PGP*, pe calculatorul lui A, mesajul (P) este codificat prin dispersie (hash), utilizând *MD5*. Codul rezultat este criptat utilizând cheia RSA privata a utilizatorului A (DA). Apoi, codul de dispersie criptat, împreuna cu mesajul original, sunt concatenate într-un singur mesaj *P1*. Acest mesaj este comprimat în format *ZIP*, rezultând mesajul *P1.ZIP*.

În acest stadiu, programul *PGP* cere utilizatorului A sa introduca un sir de caractere orisicare. Viteza de tastare a caracterelor sirului si sirul propriu zis sunt utilizate pentru a genera o cheie de mesaj de tip *IDEA* de 128 biti (KM). Cheia *KM* este utilizata în continuare pentru a cripta *P1.ZIP* cu *IDEA* prin metoda de tip reactie cifrata. În plus, *KM* este criptat cu ajutorul cheii publice a lui B (*EB*). Aceste doua componente sunt apoi concatenate, iar rezultatul, convertit în baza 64. Mesajul rezultat va contine litere, cifre si simboluri care pot fi usor transmise prin Internet, cu o probabilitate maxima de a ajunge nealterat la destinatie.

Procesul invers, de decriptare decurge dupa cum urmeaza: mesajul receptionat este mai întâi reconvertit din baza 64 si decriptat cu cheia *RSA* privata. În urma acestei decriptari se obtine mesajul *P1.ZIP*. Dupa decomprimarea acestuia, se separa textul simplu de codul cifrat, care mai apoi se decripteaza utilizând cheia publica a lui A. Daca codul textului clar este identic cu calculul facut cu ajutorul lui MD5, se poate spune cu siguranta ca mesajul este corect si ca provine de la A.

PEM- Privacy Enhanced Mail (posta cu confidentialitate sporita)

Spre deosebire de *PGP*, care nu este (înca) recunoscut ca un standard Internet, *PEM* este considerat, oficial, standard Internet. Mesajele criptate folosind *PEM* sunt mai întâi transformate într-o forma canonica, apoi este calculat un cod de dispersie (folosind MD5 sau MD2). Dupa aceasta urmeaza o concatenare a codului rezultat cu mesajul initial, urmând o criptare folosind *DES* cu o cheie pe *56* de biti. Utilizarea unei chei pe, doar, *56* de biti este "batator la ochi" si ne duce cu gândul la initiativa *PGP* de a evita algoritmi "certificati" de anumite organizatii guvernamentale.